Knowledge Discovery Engine

Inventor: Ben A. Hitt

Background of the Invention

Field of the Invention

This invention relates to pattern matching, and more specifically, to a method for recognizing and classifying biological states based on bio-markers produced by high throughput bio-assay techniques by combining an evolutionary computing algorithm, an adaptive pattern recognition component, and a means for determining cluster homogeneity.

Summary of the Invention

The present invention solves the problems associated with conventional methods of identifying, matching, and categorizing biological states based on bio-markers produced by high throughput bio-assay techniques by using a knowledge discovery engine (KDE) that combines three cooperating algorithmic subsystems: an evolutionary computer algorithm, an adaptive pattern recognition component, and a means for determining cluster homogeneity. The KDE of the present invention is capable of recognizing and classifying protein markers or patterns, called bio-markers, produced by high throughput bio-assay methods. These bio-markers reflect specific biological states either by the pattern or by the number of occurrences found. The identification of one or more bio-markers identifies certain medical conditions, e.g., cancer or other diseases.

The principle advantage to the KDE of the present invention is the unique combination of using three different algorithmic subsystems in identifying and classifying patterns embedded in a data string. Further, the present invention requires less memory during operation and performs pattern matching with a higher degree of accuracy and in less time. For example, in a test set

comprising 20 samples of chromosome strings, the KDE of the present invention identified those samples having breast cancer while making only two false positive readings. Most importantly, the KDE completed its evaluation of the test set within ONE hour.

Brief Description of the Figures

The present invention is described with reference to the accompanying drawings. In the drawings, like reference numbers indicate identical or functionally similar elements. Additionally, the left-most digit(s) of a reference number identifies the drawing in which the reference number first appears.

FIG. 1 is a block diagram showing an exemplary computer system useful for implementing the present invention;

FIG. 2 is a block data diagram showing the data flow for a knowledge discovery engine of the present invention;

FIG. 3 is a control flow diagram showing the top level operation of the knowledge discovery engine;

FIG. 4 is a control flow diagram showing the processing of chromosome strings using a genetic algorithm;

FIG. 5 is a control flow diagram showing the creating of a lead cluster map for each processed chromosome string;

FIG. 6 is a control flow diagram showing the computing of a variance across all clusters in a lead cluster map; and

FIG. 7 is a control flow diagram showing the reprocessing of processed chromosome strings using the genetic algorithm.

Detailed Description of the Preferred Embodiments

1. Host system of a preferred environment for the present invention

The chosen embodiment of the present invention is computer software executing within a computer system. FIG. 1 shows an exemplary computer system. The computer system 102 includes one or more processors, such as a processor 104. The processor 104 is connected to a communication bus 106.

The computer system 102 also includes a main memory 108, preferably random access memory (RAM), and a secondary memory 110. The secondary memory 110 includes, for example, a hard disk drive 112 and/or a removable storage drive 114, representing a floppy disk drive, a magnetic tape drive, a compact disk drive, a program cartridge and cartridge interface (such as that found in video game devices), a removable memory chip (such as EPROM, or PROM), etc. which is read by and written to by a removable storage unit 116. The removable storage unit 116, also called a program storage device or a computer program product, represents a floppy disk, magnetic tape, compact disk, etc. As will be appreciated, the removable storage unit 116 includes a computer usable storage medium having stored therein computer software and/or data. The removable storage drive 114 reads from and/or writes to a removable storage unit 116 in a well known manner.

The computer system 102 may also include other similar means for allowing computer programs or other instructions to be loaded. Such means can include, for example, a communications interface 118. Communications interface 118 allows software and data to be transferred between computer system 102 and external devices. Examples of communications interface 118 can include a modem, a network interface (such as an Ethernet card), a communications port, etc. Software and data transferred via communications interface 118 are in

the form of signals which can be electronic, electromagnetic, optical or other signals capable of being received by communications interface 118.

In this document, the term "computer program product" is used to generally refer to removable storage unit 116, a hard disk installed in hard disk drive 112, and signals transferred via communications interface 118. These computer program products are means for providing software to a computer system 102.

In an embodiment where the invention is implemented using software, the software may be stored in main memory 108, or in a computer program product and loaded into computer system 102 using removable storage drive 114, hard disk drive 112, or communications interface 118. The software, when executed by the processor 104, causes the processor 104 to perform the functions of the invention as described herein.

In another embodiment, the invention is implemented primarily in hardware using, for example, a hardware state machine. Implementation of the hardware state machine so as to perform the functions described herein will be apparent to persons skilled in the relevant arts.

The preferred embodiment of the present invention is also directed to a computer system 102 including a display device 120 and one or more input peripherals 122. The display device 120 represents a computer screen or monitor on which a graphical user interface, including a window environment, may be displayed. The input peripherals 122 include, for example, a keyboard, a mouse, a light pen, a pressure-sensitive screen, etc., which provide a user with the capability of entering input to the computer system 102.

The preferred embodiment of the present invention is directed to execute on a computer system 102 using the UNIX operating system. UNIX is commercially available and is well known in the relevant arts. The preferred computer system 102 is an IBM compatible personal computer, but the present invention also can be developed on a workstation or mainframe computer. The present invention is described in terms of a computer system 102 having a single processor 104 for convenience purposes only. It would be readily apparent, however, to one skilled in the relevant arts to use a computer system 102 having multiple processors 104, thereby executing the present invention in parallel. There are no memory requirements for developing and executing the present

invention. However, the computer system 102 achieves better performance with more main memory 108 and secondary memory 110. The preferred embodiment of the present invention is implemented in software, and more specifically, is written in the programming language C++. The preferred embodiment is described in these terms for convenience purpose only. Other comparable computer systems 102, operating systems, and programming languages could alternatively be used.

2. Overview of a Knowledge Discovery Engine

FIG. 2 is a block diagram showing the data flow for a knowledge discovery engine (KDE) 202 of the present invention. The KDE 202 inputs three types of data: data strings, e.g., chromosome strings 204, chromosome variables 208, and a user acceptable minimum 206. The chromosome strings 204 comprises data strings, e.g, bio-marker patterns, that are to be analyzed. The user input acceptable minimum 206 defines the "fitness" needed for a match. The chromosome variables 208 define the variables that the KDE 202 will look for in each chromosome string 204. Once the KDE 202 has completed its analysis of the chromosome strings 204, it outputs its findings in a chromosome map 210.

The present invention is described in terms of chromosome strings 204 for convenience purpose only. It would be readily apparent to use comparable data strings carrying bio-markers with the present invention

The KDE 202 first analyzes each data string 204 by a genetic algorithm. The use of a genetic algorithm is for convenience purpose only and it would be readily apparent for one of ordinary skill in the relevant art to use any comparable evolutionary method. The genetic algorithm randomly selects a population of candidate data element sets. It tests each data element set for how well it segments the collection of data into meaningful groups or clusters. The KDE 202 uses conventional methods for choosing the more fit sets of data elements for reproduction, mating, and survival. The genetic algorithm creates processed chromosome strings that is better at segmenting the collection than the previous population.

The adaptive pattern recognition component is a self-organizing system that finds similar groups of records in the data collection comprised of the processed chromosome strings. The degree of similarity is based on the data elements presented to it. In the preferred embodiment, the adaptive pattern recognition component is the lead cluster map, but this is for convenience purpose only. It would be readily apparent to anyone of ordinary skill in the relevant art to use any comparable algorithm. The lead cluster map is an established program in the art of data mining and discovery. It establishes clusters of data records around centroids in high order dimensional spaces. The membership of a record to a cluster is determined by Euclidean distance. If the Euclidean distance between a centroid and the record places the record inside a decision hyper-radius, the record belongs to the cluster surrounding the centroid. If the Euclidean distance between the record and any existing centroid is greater than the decision hyper-radius, the record establishes a new centroid and a new cluster.

The means for determining cluster homogeneity is a statistical measure of the variability of records belonging to a cluster with respect to specific behaviors, outcomes, attributes or the like. In the preferred embodiment, variance is used as the measure of homogeneity, but this is for convenience. It would be readily apparent to one of ordinary skill in the relevant art to use any statistical measure.

3. Control flow of a Knowledge Discovery Engine

FIGs. 3-7 are control flow diagrams showing the processing of data or chromosome strings 204 using a knowledge discovery engine (KBDE) 202 of the present invention. FIG. 3 is a control flow diagram showing the top level processing of the knowledge discovery engine. Processing beings at step 302 and immediately continues to step 304. In step 304, the KDE 202 processes the chromosome strings 204 using a genetic algorithm. Genetic algorithms are well known in the relevant art and are commercially available. In the preferred embodiment, the KDE 202 uses the genetic algorithm libraries called PGAPack available from Argonne National Laboratories.

The genetic algorithm inputs the chromosome strings 204 and for each data string, identifies the chromosome variables contained within the chromosome string 204. For example, for a chromosome string 204, the DNA sequence is divided into sub-strings and are analyzed for base triplets and sequences of base triplets. Then, the sub-strings are further divided and analyzed according to well-known methods for genetic algorithms, thereby creating a processed chromosome string. The KDE 202 records each chromosome variable 208 identified in a processed chromosome string in an internal database called string/cluster database 310.

Once each chromosome string 204 has been processed, the KDE 202 tests each processed chromosome string for fitness. The KDE 202 continues to step 306 and creates a lead cluster map, or grouping, for each processed chromosome string by using a pre-defined set of variables. Lead cluster mapping is well known in the relevant arts. All data regarding the lead cluster mapping of the processed chromosome strings is recorded in the string/cluster database 310.

The KDE 202 continues to step 308 wherein for each lead cluster map, it computes a variance across all of the clusters contained within that lead cluster map and records the variance in the string/cluster database 310. This step determines how homogeneous a given chromosome string 204 is to a predefined set of chromosome variable. Upon completion of step 308, the KDE 202 determines a best lead cluster map; that is, it determines which lead cluster map is the "best fit" with the given set of chromosome variables.

The KDE 202 continues to step 314 to determine whether the best lead cluster map is less than an acceptable minimum. The acceptable minimum may either be input by the user, or pre-defined within the KDE 202. If step 314 determines that the best lead cluster map is less than the acceptable minimum, then processing proceeds to step 316. In step 316, the KDE 202 records its final mapping in a chromosome map 210 and displays the best lead cluster map along with the matching variables. For example, the KDE 202 would display that a specific chromosome string contains those protein patterns matching breast cancer. After displaying the results for each chromosome string 204, the KDE 202 continues to step 206 and ends its processing.

Returning to step 314, if the KDE 202 determines that the best lead cluster map is not less than the acceptable minimum, the KDE 202 proceeds to step 312. In step 312, the KDE 202 re-

processes each processed chromosome string using the genetic algorithm. The generic algorithm inputs the data for each processed chromosome strings from the string/cluster database 310 and re-analyzes them according to the last set of information. After completing the re-ranking of the processed chromosome strings, the KDE 202 returns to step 306 to create new lead cluster maps for each processed chromosome string. The processing continues as described above.

FIG. 4 is a control flow diagram showing the processing of chromosome strings 204 using a genetic algorithm. Processing begins at step 402 and immediately continues to step 404. In step 404, the KDE 202 determines whether any chromosome strings 204 remain that have not been processed by the genetic algorithm. If no chromosome strings 204 remain unprocessed, the KDE 202 proceeds to step 408 and returns to step 304, thereby continuing immediately to step 306. If a chromosome string 204 remains that has not been processed by the genetic algorithm, the KDE 202 proceeds to step 406 wherein it processes the chromosome string 204 with the genetic algorithm, thereby creating a processed chromosome string. The KDE 202 records its analysis of the chromosome string 204 in the string/cluster database 310. After processing the chromosome string 204, the KDE 202 returns to step 404 which is described in detail above.

FIG. 5 is a control flow diagram showing the creating of a lead cluster map for each processed chromosome string. Processing begins at step 502 and immediately continues to step 504. In step 504, the KDE 202 determines whether any processed chromosome strings remain for which the KDE 202 has not created a lead cluster map. If no processed chromosome strings remain, the KDE 202 proceeds to step 508 and returns to step 306, thereby continuing immediately to step 308. If a processed chromosome string remains for which a lead cluster map has not been created, the KDE 202 proceeds to step 506 wherein it creates a lead cluster map and records the lead cluster map 204 in the string/cluster database 310. After processing the processed chromosome string, the KDE 202 returns to step 504 which is described in detail above.

FIG. 6 is a control flow diagram showing the computing of a variance across all clusters in a lead cluster map. Processing begins at step 602 and immediately continues to step 604. In step 604, the KDE 202 determines whether any lead cluster maps remain that the KDE 202 has not calculated a variance. If no lead cluster map remain unprocessed, the KDE 202 proceeds to step 608

and returns to step 308, thereby continuing immediately to step 314. If a lead cluster map remains for which a variance has not been calculated, the KDE 202 proceeds to step 606 wherein it processes the lead cluster map according to conventional methods. The KDE 202 records the calculated variance for the lead cluster map in the string/cluster database 310. After processing the lead cluster map, the KDE 202 returns to step 604 which is described in detail above.

FIG. 7 is a control flow diagram showing the reprocessing of processed chromosome strings using the genetic algorithm. Processing begins at step 702 and immediately continues to step 704. In step 704, the KDE 202 determines whether any processed chromosome strings remain in the string/cluster database 310 that have not been re-processed by the genetic algorithm. If no processed chromosome strings remain unprocessed, the KDE 202 proceeds to step 708 and returns to step 312, thereby continuing immediately to step 306. If a processed chromosome string remains that has not been re-processed by the genetic algorithm, the KDE 202 proceeds to step 706 wherein it re-processes the processed chromosome string with the genetic algorithm. The KDE 202 records its new analysis of the processed chromosome string in the string/cluster database 310. After re-processing the processed chromosome string, the KDE 202 returns to step 704 which is described in detail above.

Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not limitation. It will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the spirit and scope of the invention as defined in the appended claims. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.